算術演算を行う量子回路の構成

AUTHOR(S):

國廣, 昇

CITATION:

國廣, 昇. 算術演算を行う量子回路の構成. 数理解析研究所講究録 2003, 1335: 127-134

ISSUE DATE:

2003-07

URL:

http://hdl.handle.net/2433/43348

数理解析研究所講究録 1335 巻 2003 年 127-134

算術演算を行う量子回路の構成

國廣 昇

NOBORU KUNIHIRO

電気通信大学 情報通信工学科

THE UNIVERISTY OF ELECTRO-COMMUNICATIONS*

Abstract

1994 年に，P. Shor は量子計算機の実現を仮定すると素因数分解問題，離散対数問題が多項式時間で解くことが出来ることを明らかにした．その結果，RSA 暗号，ElGamal 暗号といった現在良く使われている暗号が破られることが明らかになっている．Shor のアルゴリズムは，主にべき乗剰余演算と量子フーリエ変換により構成されているが，本発表では，べき乗剰余計算を効率的に行う量子回路について検討を行う．公開鍵暗号の暗号化等において用いられている右向き binary method，Montgomery Reduction といった技法が，量子回路を構成する場合でも適用できることを確認する．

また，べき乗剰余を始めとする算術演算を効率的に実行する量子回路に関しては，これまであまり研究されてこなかったという事実がある．本研究は，算術演算を行う量子回路の構成を行う上での足がかりの研究にもなっている．

1 Introduction

実質上の標準の暗号である RSA 暗号 [1] の安全性は，大きな合成数の素因数分解が難しいことに安全性の根拠を置いている．1994 年，Shor [2] は量子計算機を用いれば，素因数分解が多項式時間で可能であることを示し，量子計算機が実現すれば，RSA 暗号も破られることを示した．しかし，Shor の示した結果は直ちに，RSA 暗号の崩壊を意味するものではない．量子計算機は，実現までかなりの時間が必要と考えられているためである．

量子計算機実現に向けて，乗り越えなくてはならない課題は大きく分けて二種類ある．一つは，「量子計算機の物理的実現」である．量子計算機の物理的実現へ向けた実験が数多く行われている．しかし，これまでに実用的なレベルで構成された例はなく，核磁気共鳴 (NMR) を用いた Chuang らのグループの実験 [3] で実現された 7qubit が最高である．

もう一つは，「量子計算機に適した回路理論の構成」である．量子計算機は，可逆な回路で構成されなくてはならないため，古典回路とは異なった特徴を持つ．すなわち，量子回路では AND や OR と言った非可逆な回路は使えず，制御 NOT や Toffoli Gate 等の可逆な回路で構成しなくてはならない．そのため，古典回路とは違った，回路構成の困難さがある．もっとも，Bernstein and Vazirani [4], Yao [5] の結果によって，Deterministic Turing Machine (DTM) により多項式時間で計算されることは，高々多項式時間の速度低下で，量子回路により計算されることがわかっている．しかしながら，一般的に DTM から変換された量子回路は十分に効率的であるとは言えないため，量子回路の機構に適した回路構成が望ましい．

Shor の素因数分解アルゴリズムの動作原理を振り返る．素因数分解したい数を n とし，$a \in Z_n$ とする．この時，関数 $f(x) = a^x \bmod n$ の周期を量子計算機により求め，その周期を基にして，古典的に素因数分解を求めている．Shor の素因数分解アルゴリズムにおいて，最も時間の要する部分は，べき乗剰余演算回路である．べき乗剰余演算は，合成数 n，被べき乗数 a，べき乗数 x とした時に，$a^x \bmod n$ を求める演算である．この演算は，並列化が困難で，量子フーリエ変換といった Shor のアルゴリズムの他の部分よりも本質的に時間が必要であることが知られている．

Shor のアルゴリズムの提案後，べき乗剰余演算を行う方式がいくつか提案されている [6, 7]．その方式は主に，必要となる qubit 数を少なくすることを目的としている．例えば，Beauregard [7] は，素因数分解したい数 n の bit 長を N とすると，$2N+3$ qubit でべき乗剰余を実現する回路を提案している．現在 7qubit までしか実現されていない状況を見ると，より少ない qubit で回路を実現することは，極めて重

*kunihiro@ice.uec.ac.jp

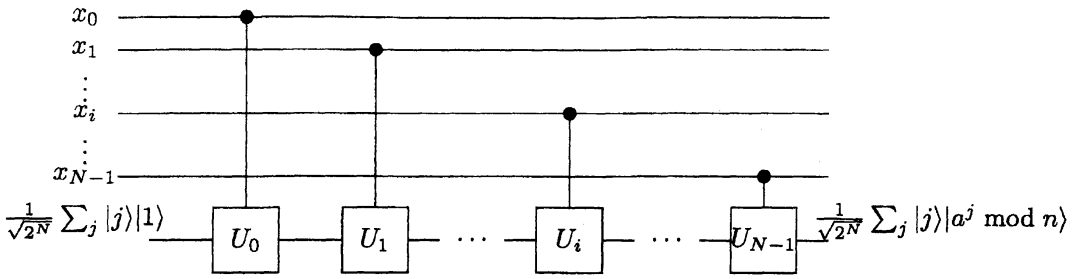

図 1: 左向き binary method

要である．しかし，量子計算機の実現を妨げている要因は，qubit 数を大きくすることができないからだけではない．他の大きな要因は，デコヒーレントまでの時間が短いため，すぐに状態が乱れてしまうことである．これは，計算時間を短くしなくてはならないことを意味する．一般に，より短い qubit で回路を構成すると，より多くの計算時間を必要するため，両立することは難しい．例えば，前述の Beauregard の回路は，量子フーリエ変換を繰り返し用いるため，計算時間は $O(N^3 \log N)$ 必要となる．$3N$ qubit 必要とする回路を用いた場合の計算量は，$O(N^3)$ であることを考慮すると，Beauregard の方式は効率的でない．

さらに，量子回路固有の問題として，回路構成の容易さが重要となる．出現する基本回路の種類が多くなると，物理的実現がより困難となる．すなわち，出現する基本回路の種類が少ない回路の構成が不可欠となる．

本研究では，必要とする qubit は比較的多いが，計算時間が短い量子べき乗剰余回路を提案する．Montgomery Reduction [8] と右向き binary method [9] を組み合わせることにより，回路を構成している．Montgomery Reduction は，m を適当に選んだ自然数として，$\bmod 2^m$ の演算により，剰余演算を行い，$\bmod n$ での演算を排除する．これにより，計算時間の短縮につなげる．また，右向き Binary Method を採用することにより，出現する回路の種類を少なくすることに成功している．

べき乗剰余演算だけでなく，より一般の算術演算を効率的に行う量子回路の構成も重要な研究課題である．Euclid の互除法等の算術演算は，量子アルゴリズムにおいても活用される可能性があるためである．より一般の算術演算を行う量子回路の構成手法に関する研究の足がかりとしても，べき乗剰余演算の構成は重要である．

2 従来の方法

Nielsen-Chuang [10] の教科書に記述されている Shor のアルゴリズム，特に，べき乗剰余演算に関して振り返り，これまで行われてきた研究を述べる．

2.1 べき乗剰余アルゴリズム

べき乗剰余演算とは，n, x を自然数，$a \in Z_n$ とした時に，$a^x \bmod n$ を計算する演算である．
以下のユニタリ変換の集合 $\{U_i\}_{i=0}^{N-1}$ を考える．ただし，N を n のビット長とする．

$$U_i|y\rangle = |ya^{2^i} \bmod n\rangle.$$

これらのユニタリ変換を用いることにより，べき乗剰余演算は図 1 のように構成される．図 1 において，黒丸は通常，制御ビットと呼ばれている．制御ビット x_i が，1 の時には，最終レジスタにはユニタリ変換 U_i が施され，0 の時には，何も施されない．この演算は制御 U_i 演算と呼ばれる

図 1 の回路により正しくべき乗剰余演算が行われるのは，以下の恒等式による．

$$\begin{aligned} a^x &= (a^{2^{N-1}})^{x_{N-1}} \times (a^{2^{N-2}})^{x_{N-2}} \times \cdots (a^{2^0})^{x_0} \\ &= (v_0)^{x_0} \times (v_1)^{x_1} \times \cdots \times (v_{N-1})^{x_{N-1}}, \end{aligned}$$

ただし，$x = x_{N-1}x_{N-2}\cdots x_1x_0$ であり，$v_i = a^{2^i} \bmod n$ である．

このべき乗剰余演算は，左向き binary method と呼ばれる方式に対応する．x の二進系列を $x_0, x_1, \ldots, x_{N-1}$ と順に左向きに処理しているからである．

注意 1 図 1 の量子回路に関していえば，左向きに x_i を処理しても，右向きに処理をしても結果，効率とも同じである．量子回路で構成する場合は，N 個のユニタリ変換 $\{U_i\}$ を事前に構成するため，どちら向きに計算をしても同じになる．古典の場合は，事前に v_i を求めることはなく，計算をしていく段階で順次 $v_i = {v_{i-1}}^2 \bmod n$ と求めていくことから，左向きに処理していく必要がある．

左向き binary method を採用する際，問題となるのは，N 種類のユニタリ変換 U_i を構成する必要がある点である．将来的には，どのような回路でも容易に構成できるような状況になるかもしれないが，そのような場合には，上記の問題は些細なことになる．しかし現在の技術では，より少ない種類の基本回路で，構成できたほうが望ましいと考えられる．

注意 2 回路は最終的には，制御 NOT や Toffoli Gate まで分解することは可能である．しかし，この発表中では，回路の構成要素として，上記の U_i 等を考える．

2.2 modular addition

上記のアルゴリズムにおいては，modular multiplication を組み合わせることにより，べき乗剰余演算を構成している．また，一般的には，modular addition を組み合わせることにより，modular multiplication を構成する．modular addition: $a+b \bmod n$ は以下の演算である．ただし，$a, b \in Z_n$ である．

$$a+b \bmod n = \begin{cases} a+b & \text{if } a+b<n \\ a+b-n & \text{if } a+b \geq n. \end{cases}$$

この演算をより少ない qubit で実現する方式がいくつか提案されている．量子フーリエ変換を利用する quantum addition [11] を採用することにより，Beauregard [7] は $2N+3$ qubit のみでべき乗剰余演算を成功している．ただし，このアルゴリズムは，多くの計算時間を必要とし，また回路は煩雑になる．一般に，modular addition の構成を困難にしているのは，法を n とした演算を行っている点である．つまり，$a+b$ と n の大小関係により，処理を変える必要があり，処理が煩雑となっている．

3 提案回路

2 章では従来の方式には，二つの点で問題があることを見てきた．

1. 左向き binary method の採用しているため，回路の構成要素が多く必要である．
2. 法を n として演算をする必要があるため，回路が煩雑になる．

上記の問題点を解決する方式としてそれぞれ，(1) 右向き binary method の適用，(2) Montgomery Reduction の適用を試みる．(1) の右向き binary method の採用により，構成回路の種類を少なくし，(2) の Montgoemry Reduction の採用により，法を n とした演算を回避する．

3.1 右向き binary method の適用

右向き binary method を採用した場合，通常のべき乗剰余演算は，以下のアルゴリズムにより実現される．ただし，入力を $a \in Z_n$, $x = x_{N-1}x_{N-2}\cdots x_0$ とする．

初期値 $y=1, c=N-1$

Step1 $y \leftarrow y^2 \bmod n$

Step2 If $x_c = 1$, then $y \leftarrow y * a \bmod n$.

Step3 $c = c-1$; Step 1 へ.

x_i を最後まで読みきると終了し，y を出力する．上のアルゴリズムによりべき乗剰余演算が行われるのは，以下の恒等式による．

$$a^x = ((((((a^{x_{N-1}})^2 \times a^{x_{N-2}})^2 \times a^{x_{N-3}})^2)\cdots)^2 \times a^{x_0}.$$

このアルゴリズムに基づく，べき乗剰余演算を行う量子回路を構成する．この量子回路は，2 乗剰余演算 (Step1) と制御 a 倍剰余演算 (Step2) の二種類により構成される．図 2 に回路を記述する．回路中，SS は 2 乗剰余回路，M_a は a 倍剰余回路である．左向き binary method の場合は，N 種類の制御 a^{2^i} 倍剰余演算を構成する必要があったが，右向きの場合は構成すべき回路の種類が少なくなっている．

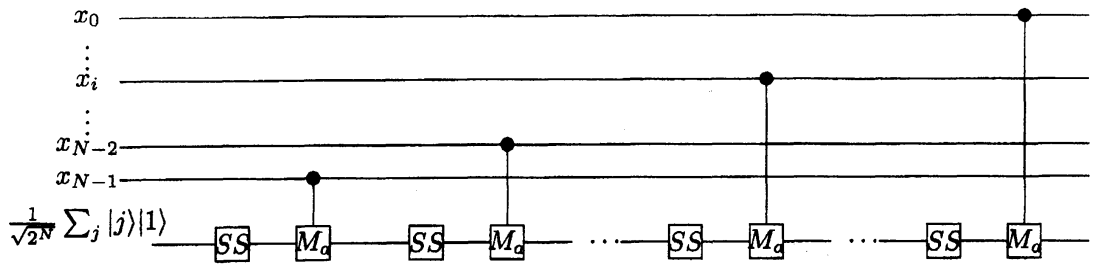

図 2: 右向き binary method

3.2 Montgomery Reduction の適用

Montgomery Reduction [8] は次のように定式化される．入力を n^2 程度の自然数 Y として，R を $n < R = 2^m$ となるように適当に選んだ数とする．Y に対する Montgomery Reduction $MR(Y)$ を $Y \cdot R^{-1} \bmod n$ と定義する．$MR(Y)$ は，以下のアルゴリズムにより計算される．ただし，前計算として，$V = -n^{-1} \bmod R$ を事前に計算しておく．$MR(Y)$ を求めるのに法を n とした演算をする必要がないことが重要である．

Input: Y

Output: $MR(Y) = Y \cdot R^{-1} \bmod n$

Step1: $W_1 \leftarrow Y \cdot V \bmod 2^m$

Step2: $W_2 \leftarrow Y + W_1 \cdot n$

Step3: $MR(Y) \leftarrow W_2/2^m$

このアルゴリズムにより，正しい値が返されることは容易に確認できる．$MR(Y)$ を求めるのに，法を 2^m とした剰余演算と 2^m による割り算しか必要としない点が重要である．

また，Montgomery Reduction には，別の計算法がある．Kaliski らの改良 [12] から，以下のアルゴリズムは直接導き出される．

Input: Y

Output: $MR(Y) = Y \cdot R^{-1} \bmod n$

Step1: 以下の処理を m 回繰り返す．

Step1-1: Y が奇数の時，$Y \leftarrow Y + n$

Step1-2: $Y \leftarrow Y/2$

Step2: $MR(Y) \leftarrow Y$ として出力．

このアルゴリズムにより，正しい値が返されることは容易に確認できる．この演算は，n の条件付加算とビットシフトのみで構成されることに注意せよ．

一般に，Montgomery Reduction は 1 対 1 の関数ではない．そのため，ユニタリ変換を U_{MR} すると，

$$|Y\rangle \xrightarrow{U_{MR}} |MR(Y)\rangle|garbage\rangle$$

となる．ただし，単純な 2 乗演算を行うユニタリ変換を U_S とすると，

$$|Y\rangle|0\rangle \xrightarrow{U_S} |Y^2\rangle \xrightarrow{U_{MR}} |MR(Y^2)\rangle|0\rangle$$

とすることができる．$U_1 = U_{MR} \circ U_S$ とすると，$|Y\rangle \xrightarrow{U_1} |MR(Y^2)\rangle$ となる．また，X 倍乗算ユニタリ回路を U_X とすると，

$$|Y\rangle|0\rangle \xrightarrow{U_X} |XY\rangle \xrightarrow{U_{MR}} |MR(XY)\rangle|0\rangle$$

とすることができる．$U_2 = U_{MR} \circ U_X$ とすると，$|Y\rangle \xrightarrow{U_2} |MR(XY)\rangle$ となる．

Montgomery Reduction の逆演算 (lifting) を考える．つまり，$A^{(R)} = A \cdot R \bmod n$ とする．この時，$(AB)^{(R)}$ は，以下のように計算される．

$$\begin{aligned}(AB)^{(R)} &= ABR \bmod n = A^{(R)} \cdot B^{(R)} \cdot R^{-1} \\ &= MR(A^{(R)} \cdot B^{(R)}).\end{aligned}$$

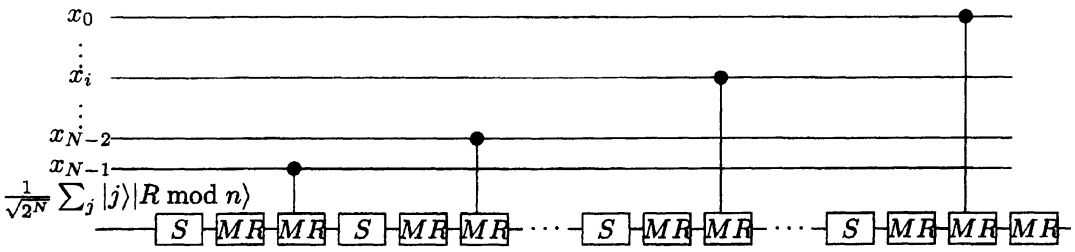

図 3: 提案する回路

つまり，ある数 AB の Montgomery lift を求めるには，A, B の Montgomery lift をかけたものに，Montgomery Reduction を施せばよい．以上の考察より，$a^x \bmod n$ を求めるには，以下の手順を踏めば良い．

1. a の Montgomery lift $a^{(R)} = a \cdot R \bmod n$ を求める．
2. 右向き binary method を用いて，$(a^x)^{(R)}$ を求める．
3. $(a^x)^{(R)}$ を Montgomery Reduction をし，$a^x \bmod n$ を求める．

$a^{(R)}$ を用いて $(a^x)^{(R)}$ を右向き binary method により以下の手順で計算する．

初期値 $y = R \bmod n$, $c = N - 1$

Step1 $y \leftarrow MR(y^2)$

Step2 If $x_c = 1$, then $y \leftarrow MR(y * a^{(R)})$.

Step3 $c = c - 1$; Step 1 へ．

さらに，$a^{(R)} = 1$ となるように a を 設定すれば，Step2 の then 以下は，$y \leftarrow MR(y)$ とすることができる．

注意 3 一般に $a^{(R)} = 1$ は成立しないが，素因数分解を行い場合は，a を一つに固定しても，ほとんどの場合，問題はない．

3.3 提案する量子回路

図 3 に，入力を $\frac{1}{\sqrt{2^N}}\sum_{i=0}^{2^N-1}|i\rangle|R \bmod n\rangle$ として，出力を $\frac{1}{\sqrt{2^N}}\sum_{i=0}^{2^N-1}|i\rangle|a^i \bmod n\rangle$ とする量子回路を記述する．ただし，a は $a^{(R)} = 1$ を満たしているとする．つまり，$a = R^{-1} \bmod n$ とする．図 3 の回路は，2 乗演算回路 S と Montgomery Reduction 回路 MR（および，制御 Montgomery Reduction 回路）により構成されている．左向き binary method を採用したときは，N 種類の回路 U_i を構成する必要があったが，右向きを採用することにより，少ない種類の回路でべき乗剰余演算を構成することに成功している．

3.4 提案法の考察

提案する回路に関して，考察を加える．

最後の MR 回路は省略化
Shor の素因数分解回路において，重要なのは，$a^x \bmod n$ の値そのものではなく，値の重ねあわせ状態である．最後の回路 MR は，Montgomery lift した値から，最終的な値を求めるだけであるので，最後の Montgomery Reduction を実行することなく，素因数分解を行うのに必要な重ね合わせ状態を得ていることになる．

最初の S, MR 回路は省略化
状態 $|R \bmod n\rangle$ に対して，順に S, MR 回路を実行すると，状態は $|MR(S(R))\rangle = |R \bmod n\rangle$ となり，変化しない．そのため，最初の S, MR は省略しても良い．

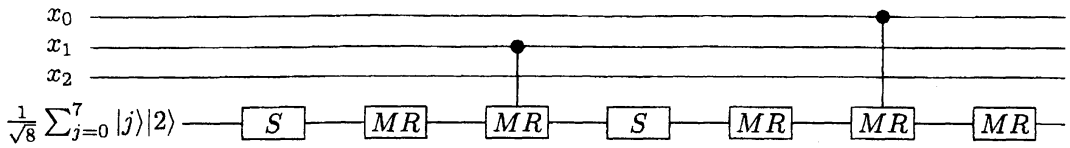

図 4: $n = 15$ の時の回路（その 1）

回路の種類の数について
提案法では，出現する回路は二乗回路 S と Montgomery Reduction 回路 MR の二種類である．それに付加して，制御 MR 回路が必要である．この回路の，制御 bit は，全て異なるため，量子計算機の実現形態によっては，異なる回路として構成しなくてはいけないかもしれない．その場合は，必要な回路の種類は，$N+2$ 種類であり，従来法と比較しての優位性はなくなる．逆に，同一の回路とみなせる場合には，提案手法は優位になる．

3.5 簡単な例 1

素因数分解したい数を $n = 15$ とした場合の回路の構成例を記述する．

まず，$R = 2^5 = 32$ と設定する．この場合，$a = (32)^{-1} \bmod 15 = 8$，$V = -15^{-1} \bmod 32 = 17$ となる．べき乗剰余回路の初期値は，$\frac{1}{\sqrt{2^3}}\sum_{i=0}^{7}|i\rangle|R \bmod n\rangle = \frac{1}{\sqrt{2^3}}\sum_{i=0}^{7}|i\rangle|2\rangle$ となる．

$n = 15, R = 32$ の時には，Montgomery Reduction: MR は次のように計算される．

Step1: $W_1 \leftarrow 17Y \bmod 2^5$

Step2: $W_2 \leftarrow Y + 15W_1$

Step 3: $MR(Y) \leftarrow W_2/2^5$

全体の回路は図 4 のようになる．すなわち，基本回路 ($S - MR -$ 制御 MR) に対して，制御ビットを x_1, x_0 と変化させて順に適用し，最後に MR を適用する．

図 4 の回路により，順に以下のように計算される．ただし，図 4 の回路における最初の S, MR に関しては省略している．

$$\begin{aligned}
\frac{1}{\sqrt{8}}\sum_{i=0}^{7}|i\rangle|2\rangle &\rightarrow \frac{1}{\sqrt{8}}\sum_{i=*0*}|i\rangle|2\rangle + \frac{1}{\sqrt{8}}\sum_{i=*1*}|i\rangle|1\rangle \rightarrow \frac{1}{\sqrt{8}}\sum_{i=*0*}|i\rangle|4\rangle + \frac{1}{\sqrt{8}}\sum_{i=*1*}|i\rangle|1\rangle \\
&\rightarrow \frac{1}{\sqrt{8}}\sum_{i=*0*}|i\rangle|2\rangle + \frac{1}{\sqrt{8}}\sum_{i=*1*}|i\rangle|8\rangle \\
&\rightarrow \frac{1}{\sqrt{8}}\sum_{i=*00}|i\rangle|2\rangle + \frac{1}{\sqrt{8}}\sum_{i=*01}|i\rangle|1\rangle + \frac{1}{\sqrt{8}}\sum_{i=*10}|i\rangle|8\rangle + \frac{1}{\sqrt{8}}\sum_{i=*11}|i\rangle|4\rangle
\end{aligned}$$

この計算が正しく行われていることを確かめるためには，最後の結果に対して，Montgomery Reduction を施した結果が，$\frac{1}{\sqrt{8}}\sum_{i=0}^{7}|i\rangle|8^i \bmod 15\rangle$ となることより確かめられる．

3.6 $n = 15$ に特化した回路

Chuang ら [3] は，一般の合成数に通用する回路ではなく，$n = 15$ の場合にのみ有効な回路を構成している．前述のアルゴリズムを簡素化することにより，彼らとほぼ同等の回路を構成できる．

3.5 章の例で見たとおり，第二レジスタに格納される値は 2^l，ただし，$l = 0, 1, 2, 3$，のみである．まず，$MR(2^l)$ の値を考える．ただし，$l = 0, 1, 2, 3$ とする．Step1 において，$W_1 = 17 \times 2^l \bmod 2^5 = 16 * 2^l + 2^l \bmod 2^5$ より，$l = 0$ の時，$W_1 = 17$，それ以外の時は，$W_1 = 2^l$ となる．Step2 において，$W_2 = 2^l + 15W_1$ より，$l = 0$ の時は，$W_2 = 1 + 15 * 17 = 32 \times 8$，それ以外の時は，$W_2 = 2^l + 15 \times 2^l = 32 \times 2^{l-1}$ となる．つまり，$MR(2^0) = 2^3$, $MR(2^l) = 2^{l-1}$ ただし，$l = 1, 2, 3$ となる．結局，working qubit が左から順に w_3, w_2, w_1, w_0 に配置されているとすると MR は 1 ビット右シフト演算となる．ただし，一番右側に来たときには，一番左のビットに値を送ると約束する．もしくは，w_3, w_2, w_1, w_0 を時計周りに配置した状況を考えたときに，MR は，時計周りに 1 ビットシフトする演算と考えることができる．この計算機

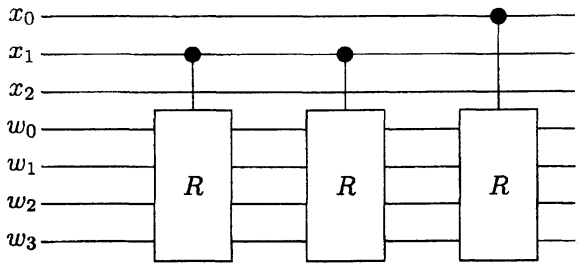

図 5: $n = 15$ の時の回路（その 2）

モデルは，Quantum Turing Machine の範疇に入らないが，量子計算機の実現形態によっては，効率的な回路となりうる．

上記の回路においては，さらに一回の二乗演算がある．この二乗演算は，制御 bit に NOT を施した後に，制御—ビット左シフトで置き換えることができる．

さらに，回路の簡素化を行うと，$n = 15$ で $a = 8$ の時の回路は図 5 のようになる．ただし，R は 1 ビット右シフトである．

注意 4 $n = 2^N - 1$ の場合にもほぼ同じ構成でべき乗剰余演算を行う回路を作ることができる．この場合，$R = 2^{N+1}, a = 2^{N-1}, V = 2^N + 1$ となる．しかしながら，この条件下での a の周期は N となるため，事前に周期はわかっており，量子計算機を用いる利点はない．$n = 2^N - 1$ の時に，量子計算機を用いて，意味のある素因数分解を求めるためには，a と 2^{N-1} 以外の異なる値を取らなくてはならない．そのため，若干の速度低下が生じる．

3.7 簡単な例 2

素因数分解したい数を $n = 91$ とした場合の回路の構成例を記述する．91 は 15 の次に素因数分解すべきターゲットと目されている数である．

まず，$R = 2^7 = 128$ と設定する．この場合，$a = (128)^{-1} \bmod 91 = 32$，$V = -91^{-1} \bmod 128 = 45$ となる．べき乗剰余回路の初期状態は，$\frac{1}{\sqrt{2^7}} \sum_{i=0}^{127} |i\rangle|37\rangle$ となる．

$n = 91, R = 128$ の時には，Montgomery Reduction: MR は次のように計算される．

Step1: $W_1 \leftarrow 45Y \bmod 2^7$

Step2: $W_2 \leftarrow Y + 91W_1$

Step3: $MR(Y) \leftarrow W_2/2^7$

実際の量子回路の次のように構成される．基本回路 ($S - MR -$ 制御 MR) に対して，制御ビットを順に $x_6, x_5, \ldots, x_1, x_0$ と変化させて適用し，最後に MR を適用する．量子計算機により，周期を求めると 12 となるはずである（もちろん，現在のところそのような量子計算機は存在しない）．$\gcd(32^6 - 1, 91) = 7$ を計算し，$91 = 7 \times 13$ という素因数分解を得る．

4 まとめ

べき乗剰余演算を行う量子回路を提案した．提案方式は，右向き binary method を採用し，Montgomery Reducton を採用するという特徴を持っている．従来の方式よりも，回路の構成要素が少ないという特徴を持っている．qubit が多く必要となるという欠点は持つが，より少ない計算時間で計算ができると期待される．

今後，Montgomery Reduction 回路等を制御 NOT により具体的に記述し，実際に必要となる qubit 数の評価，計算時間の評価を行う予定である．さらに，個の研究での知見をいかして，べき乗剰余演算以外の算術演算（Euclid の互除法等）に関しても，効率的な量子回路の構成を行う予定である．

参 考 文 献

[1] R.L. Rivest, A. Shamir and L. Adleman, "A method for obtaining digital signature and public key cryptosystems," *Comm. of ACM*, vol.21 no. 2, pp.120–126, 1978.

[2] P.W. Shor, "Algorithms for Quantum Computation: Discrete Log and Factoring," in Proc. of the 35th FOCS, pp. 124–134, 1994.

[3] L. M. K. Vandersypen, M. Steffen, G. Breyta, C. S.Yannoni, M. H. Sherwood, I. L. Chuang, "Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance," Nature 414, pp. 883–887, 2001.

[4] E. Berstein and U. Vazirani, "Quantum Complexity Theory," in Proc. of the 325h STOC, pp. 11–20, 1993.

[5] A.C. C. Yao, Quantum Circuit Complexity," in Proc. of the 34th FOCS, pp. 352–361, 1994.

[6] V. Verdal, A. Barenco, and A. Ekert, "Quantum Networks for Elementary arithmetic Operations," Phys. Rev. A, 54, pp. 147–153, 1996.

[7] S. Beauregard, "Circuit for Shor's algorithm using $2n+3$ qubits," **quant-ph/0205095v2**, 2002.

[8] P. L. Montgomery, "Modular Multiplication Without Trial Division," Mathematics of Computation, 44, 170, pp. 519–521, 1985.

[9] D. E. Knuth, "The art of Computer Programming," Addision–Wesley Publishing, 1981.

[10] M. A. Nielsen and I. L. Chuang, "Quantum Computation and Quantum Information,"Cambridge University Express, 2000.

[11] T. Draper, "Addition on quantum computer," **quant-ph/0008033**, 2000.

[12] S. Dusse and B. Kaliski Jr., "A Cryptographic Library for the Motorola DSP56000," Proc. of EUROCRYPT'90, pp. 230–243.